

ECE 445
SENIOR DESIGN LABORATORY
FINAL REPORT

CHARM
CHeap Accessible Resilient Mesh

Team No. 1

MARTIN MICHALSKI
(martinm6@illinois.edu)

MELISSA PAI
(mepai2@illinois.edu)

TREVOR WONG
(txwong2@illinois.edu)

TA: Kai Chieh (Jeff) Chang
Professor: Victor Gruev

December 7th, 2022

Abstract

This final report describes a problem we identified, our solution, and the implementation of our project for ECE 445. We provide descriptions of the project's features, design, and cost.

Contents

| | | |
|----------|-----------------------------------------------|-----------|
| 1 | Introduction | 1 |
| 1.1 | Problem | 1 |
| 1.2 | Solution | 1 |
| 1.3 | Visual Aids | 1 |
| 1.4 | High-Level Requirements | 1 |
| 2 | Design | 3 |
| 2.1 | Block Diagram | 3 |
| 2.2 | Power Management Subsystem | 3 |
| 2.2.1 | Overview | 3 |
| 2.2.2 | Requirements | 5 |
| 2.2.3 | Design Decisions | 5 |
| 2.2.4 | Power Management Subsystem RV Table | 6 |
| 2.3 | Sensor Subsystem | 7 |
| 2.3.1 | Overview | 7 |
| 2.3.2 | Requirements | 8 |
| 2.3.3 | Design Decisions | 9 |
| 2.3.4 | Sensor Subsystem RV Table | 10 |
| 2.4 | Routing Subsystem | 11 |
| 2.4.1 | Overview | 11 |
| 2.4.2 | Requirements | 11 |
| 2.4.3 | Design Decisions | 11 |
| 2.4.4 | Routing Subsystem RV Table | 12 |
| 2.5 | System Monitor Subsystem | 14 |
| 2.5.1 | Overview | 14 |
| 2.5.2 | Requirements | 14 |
| 2.5.3 | Design Decisions | 15 |
| 2.5.4 | System Monitor Subsystem RV Table | 16 |
| 2.6 | Physical Design | 17 |
| 2.7 | Tolerance Analysis | 17 |
| 3 | Cost Analysis | 18 |
| 3.1 | Parts/Materials | 18 |
| 3.2 | Estimated Hours of Development | 18 |
| 3.3 | External Materials and Resources | 18 |
| 3.4 | Approximate Total Cost | 19 |
| 4 | Conclusion | 20 |
| 4.1 | Conclusion | 20 |
| 4.2 | Ethics & Safety | 20 |
| | Appendix A Ethics and Safety | 21 |
| A.1 | Ethics | 21 |

| | | |
|-------------------|-------------------------------|-----------|
| A.2 | Safety | 21 |
| A.2.1 | Battery Safety | 22 |
| A.2.2 | User Privacy | 22 |
| A.2.3 | System Security | 23 |
| Appendix B | Review Processes | 24 |
| B.1 | Software | 24 |
| B.2 | Hardware | 24 |
| Appendix C | Soldering Verification | 25 |
| C.1 | USB-C Port | 25 |
| C.2 | Boost Converter | 25 |
| C.3 | Buck Converter | 25 |
| C.4 | Charge Controller | 25 |
| C.5 | Omega 2S+ | 26 |
| C.6 | Battery Monitor | 26 |
| C.7 | GPS Sensor | 26 |
| Appendix D | PCB Parts & Cost | 28 |
| References | | 29 |

1 Introduction

We provide details about the problem we have addressed, along with our implemented solution, in context.

1.1 Problem

Hurricanes, earthquakes, tsunamis, and other natural disasters all have the capability to destroy cellular- and inter- networks, cutting off victims from emergency services, unaffected survivors from worried loved ones, and rescuers from their command and control. This was the case in NYC after Hurricane Sandy, with a large area of the city facing cellular outages for a month after the hurricane [1], and in Japan which experienced blackouts and cellular outages after a 2011 tsunami [2].

Remote areas also suffer from limited connectivity, with connections to cabling, cell towers, and other networking hardware being few and far between [3]. In the case of an emergency, such communicative sparsity may even be fatal, e.g. a farmer which has a heart attack and is not able to call for emergency services or relatives to come to their aid.

1.2 Solution

To solve these problems, we created a set of meshing, cheap, lightweight, and self-contained wireless access points. After placement, these access points form a WiFi network, usable by rescuers, survivors, and civilians.

1.3 Visual Aids

We provide an in-context view of CHARM in Figure 1. The cloud-hosted System Monitor keeps tabs on the status of each CHARM node, with the nodes themselves being interconnected and serving their connected clients. The master router provides access to the wider internet for CHARM.

1.4 High-Level Requirements

During our demo, we met the following goals:

1. The system shall cover at least 7500 m^2 with publicly-accessible WiFi, using a maximum of 5 CHARM nodes.
2. Each node shall weigh no more than 1.5 kg and not exceed 8000 cm^3 of volume.
3. The system shall be monitored via a web-interface which is able to enumerate the location ($\pm 10 \text{ m}$), battery voltage ($\pm 0.1 \text{ V}$), and networking telemetry for each node.

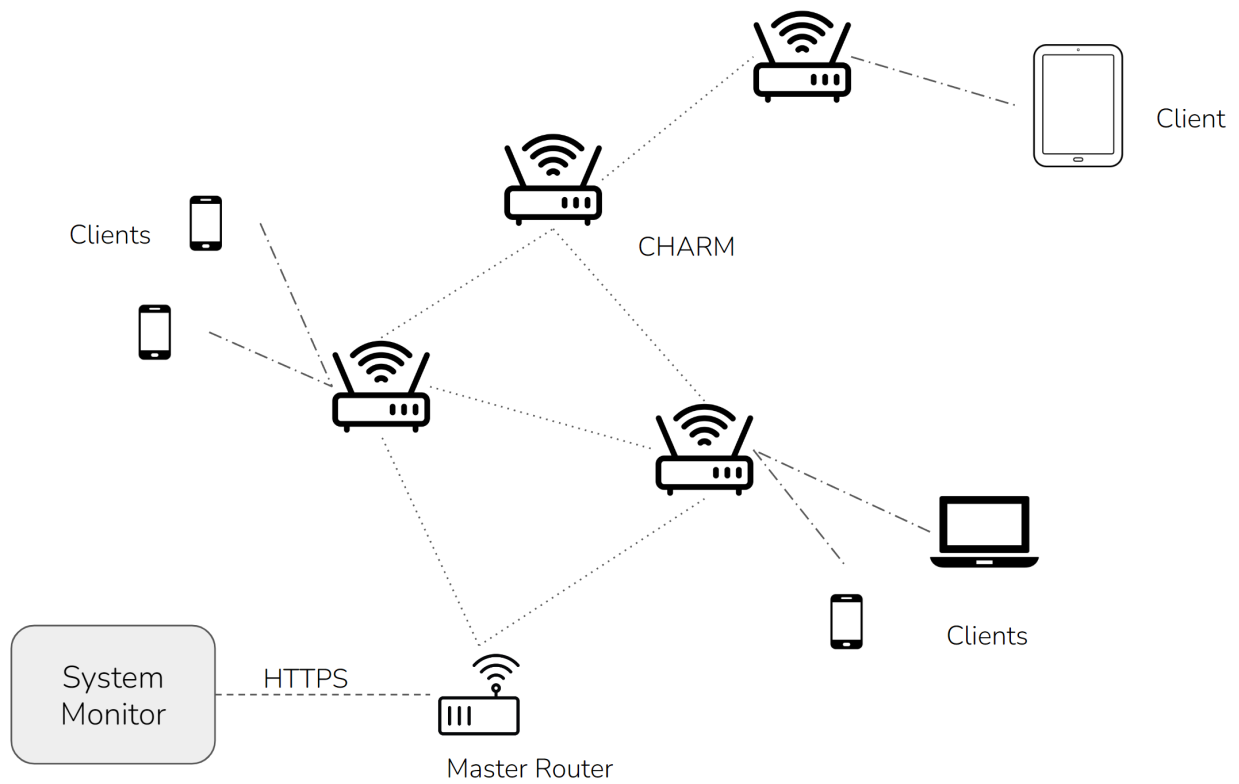


Figure 1: High-Level Overview of the CHARM System

2 Design

2.1 Block Diagram

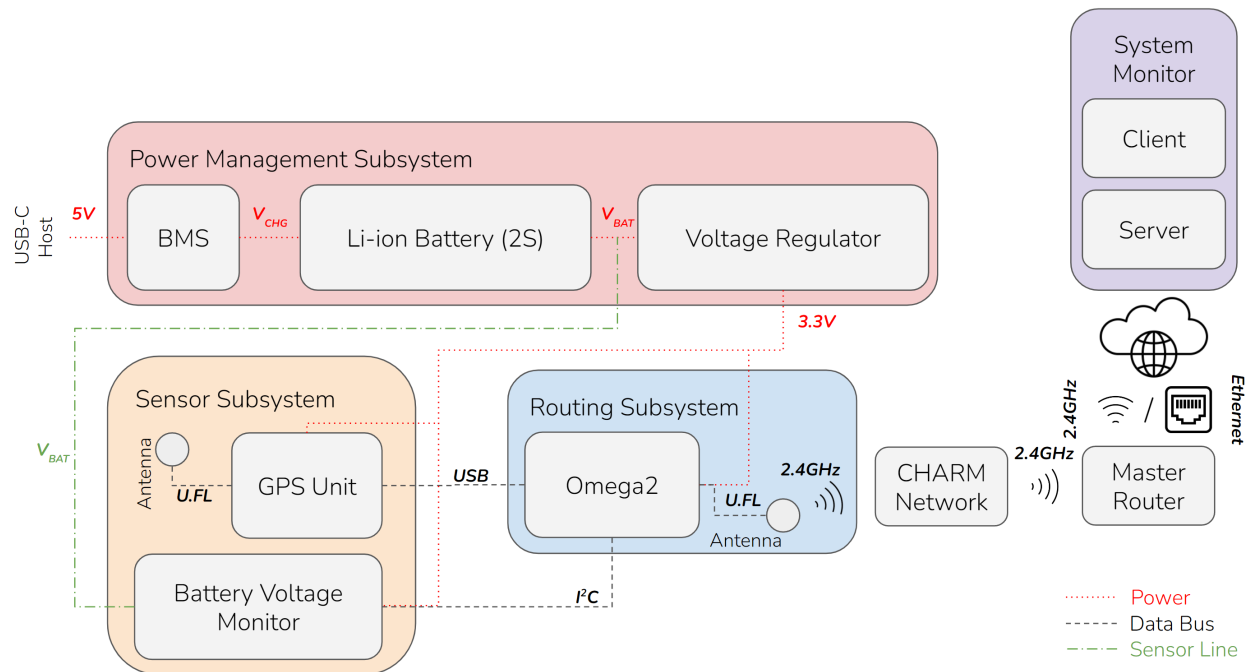


Figure 2: Block Diagram

Our design is divided into four subsystems, which are implemented in both hardware and software, and a physical enclosure. These subsystems are visible in our block diagram (Figure 2).

2.2 Power Management Subsystem

2.2.1 Overview

The power management subsystem handles charging, discharging, and voltage regulation. Specifically, it includes a USB-C port that allows the nodes to be charged with a standard USB-C charger. A boost converter and constant-current constant-voltage charging IC increase the voltage and deliver it safely to the batteries. Finally, the output of the batteries is fed through an on/off switch and into a buck converter, which reduces the voltage to power the compute module and sensors.

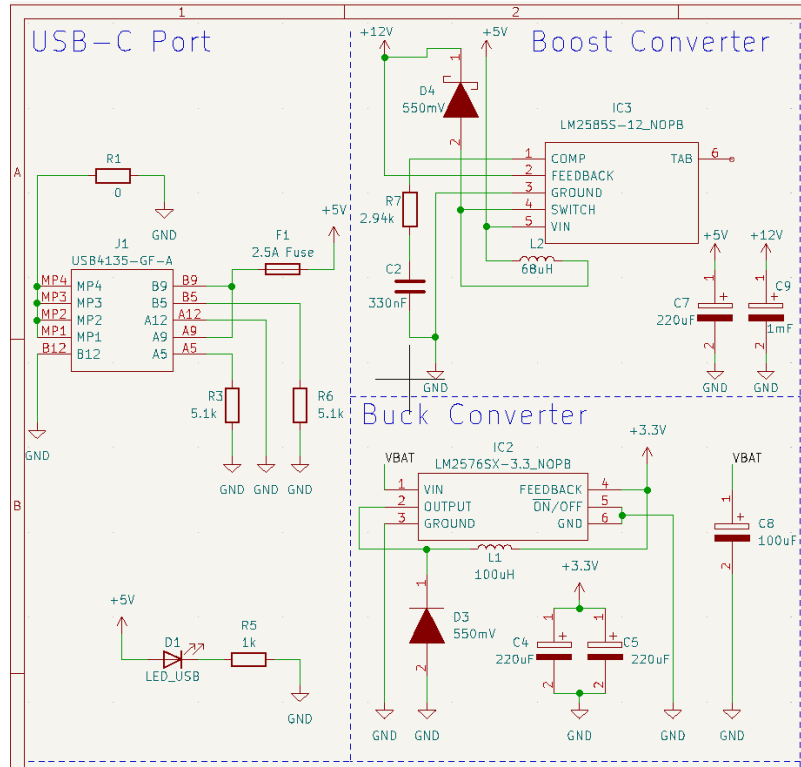


Figure 3: Schematic for Boost converter, Buck converter, and USB-C port

The USB-C port is connected with pull-down resistors to tell the connected device to supply 5 V of power on the rails, and a fuse prevents overcurrent on the input. A 0 ohm resistor is included in the design in case the USB port housing needs to be grounded, although this pad has been left unoccupied. The boost and buck converters were designed with the help of the TI Webench tool [4]. They were optimized to minimize BOM count to simplify the routing and assembly of our prototype.

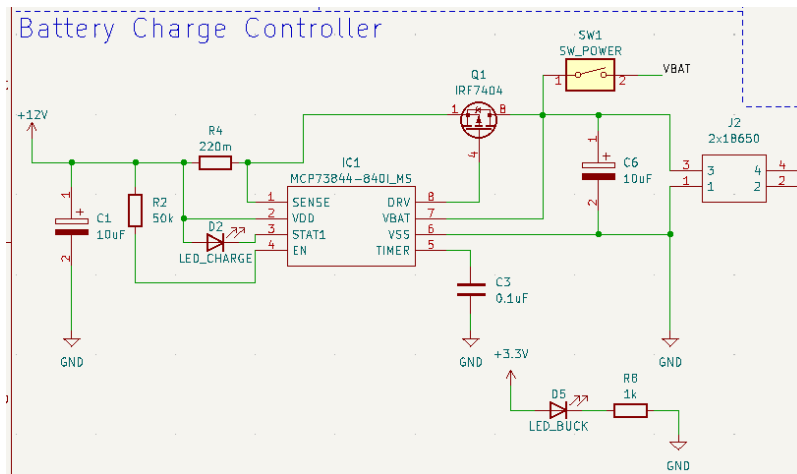


Figure 4: Schematic for Charge Controller

As suggested by a blog post [5], we used an MCP7384 to charge the 2s batteries, following its reference schematic [6]. This circuit ensures the batteries are charged in a safe and effective manner. It is important to note we use protected 18650 cells in our design, so some safeguards such as short circuit and overcurrent protection are not visible in the schematic because they are already implemented in the battery cells.

2.2.2 Requirements

The power delivery system satisfies the following requirements:

1. The system must charge 2S Li-ion battery cells with a standard USB-C charger and a peak charging current of at least 400mA.
2. The system must provide a stable 3.3 V (± 0.1 V) power source that supports up to 2 A of current draw.

These requirements were validated with procedures described in Table 1.

2.2.3 Design Decisions

We made a number of design decisions for the power subsystem. First, the boost and buck converter ICs were selected to minimize BOM count. While our implementation is more expensive than it otherwise could be, minimizing the number of passive components greatly simplifies PCB trace routing and reduces the overall footprint of our device. Additionally, using boost and buck ICs with fixed output voltages eliminates the voltage feedback resistor as a source of error.

We also selected a USB-C port for charging to make the charging process as easy and accessible for users as possible. It would have been much easier to accept a 12 V input with a barrel connector, since this would eliminate the need for a USB port and boost converter. However, these devices are not as accessible if they are not charged, and in an emergency situation a USB-C wall charger or portable charger is much more likely to be available than a 12 V power adapter with the exact barrel connector needed. Thus, we made it a priority to implement our design with USB-C, despite the increased cost and complexity.

2.2.4 Power Management Subsystem RV Table

Table 1: Power Management Subsystem RV Table. Requirements 1 & 2

| Requirements | Verification |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none">The system must charge 2S Li-ion battery cells with a standard USB-C charger and a peak charging current of at least 400mA. | <ul style="list-style-type: none">Assemble a CHARM node with a complete power delivery subsystemPlace a $10\ \Omega$ ($\pm 1\Omega$) resistor across the cathode and ground to drain the batteries to 5.2 V (± 0.1 V).Place a jumper wire between the battery pack cathode, an ammeter, and the rest of the circuit. Place a voltage sensor to measure the battery voltage against ground.Connect the battery to a USB port and begin recording the current and voltage over time.Verify the current exceeds 400 mA at least once and the voltage plotted over time is consistent with constant-current constant-voltage charging. |
| <ul style="list-style-type: none">The system must provide a stable 3.3 V (± 0.1 V) power source that supports up to 2 A of current draw. | <ul style="list-style-type: none">Assemble a CHARM node with a complete power delivery subsystem.Solder wires to the 3.3 V power rail and ground.Ensure the node is switched on, measure the voltage between the power rail and ground, and verify it is below 3.4 V.Place a 1.5Ω ($\pm 0.1\Omega$) resistor between the power rail and ground.After 1 minute, measure the voltage on the power rail and verify it is above 3.2 V. |

2.3 Sensor Subsystem

2.3.1 Overview

The sensor subsystem is responsible for collecting the GPS location of each node, monitoring the battery voltage, and communicating these metrics to the compute module.

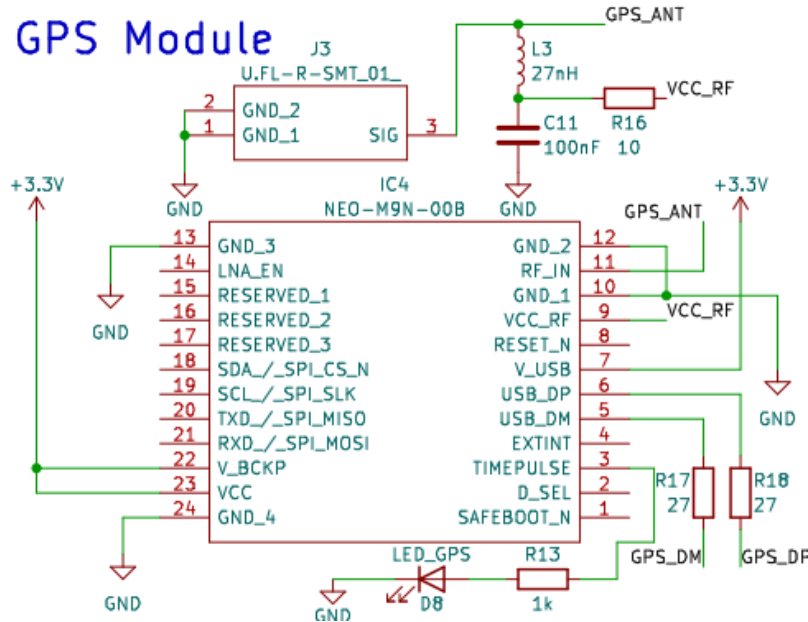


Figure 5: Schematic for GPS Module

The GPS module includes the GPS chip, an active antenna, and the necessary electrical components as suggested in the chip's (NEO-M9N) datasheet [7]. For the active antenna circuit, we followed the example circuit provided in the implementation guide[8]. This example circuit is a bias-T layout which includes a capacitor to remove noise, an inductor to isolate RF from the DC path, and a current limiting series resistor for potential short circuits.

Battery Monitor

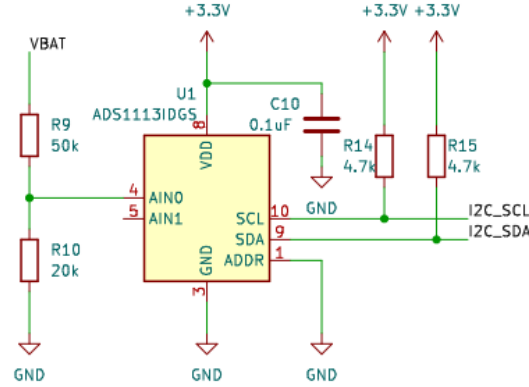


Figure 6: Schematic for Battery Monitor

The battery monitor section contains an Analog to Digital Converter that interfaces with the compute module using I^2C . A voltage divider steps the voltage down to a level that can be read by the ADC, and the I^2C wires have pull-up resistors as required by the Omega2S+ [9].

2.3.2 Requirements

1. The system must provide battery voltage measurements (± 0.1 V).
2. The system must provide a location within 10 meters of the true location when the device is deployed in an area free of large buildings or other obstacles.

These requirements were validated with procedures described in Table 2. When testing Requirement 1, we noticed that we were not getting the voltages we expected. As seen in Figure 7, the observed voltage did not align with our expectation. To resolve this issue, we created a line of best fit for V_{observed} and took it's inverse to get an equation for V_{batt} .

After this, we hooked up a power supply to the battery terminals and verified we were measuring within 0.1 V across the whole operating voltage range of our batteries.

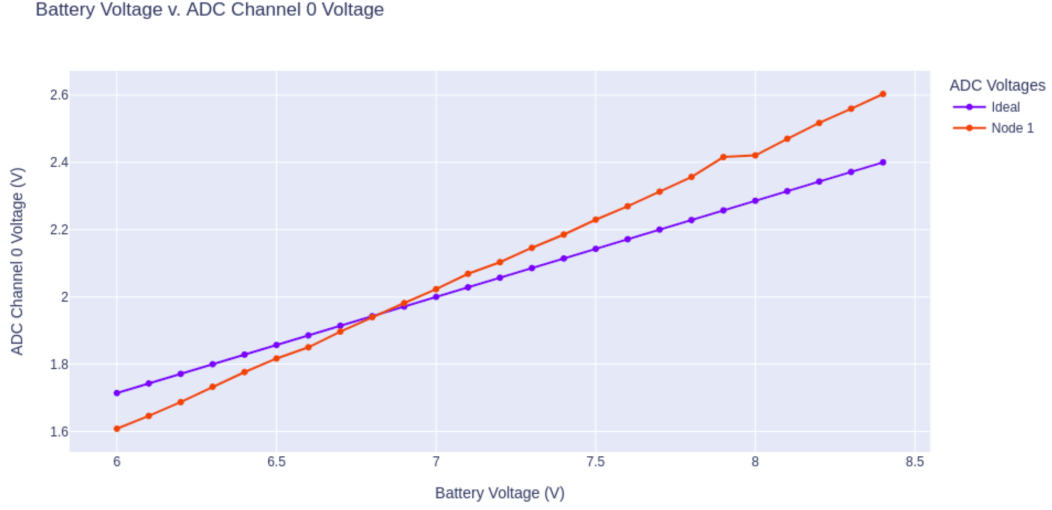


Figure 7: V_{ideal} and V_{observed}

2.3.3 Design Decisions

Our initial circuit design for GPS's communication interface used a UART port. However, after deep diving in the documentation, we realized that the all communication input ports (including UART) have internal pull-up resistors. This is not compatible with the Omega2S+, as the GPIO pins we were planning on using for UART communication must be left floating or pulled down during boot time. After carefully considering the alternatives, we ended up deciding to utilize USB to communicate to the Omega2S+, since we need to be certain the Omega2S+ will boot consistently.

2.3.4 Sensor Subsystem RV Table

Table 2: Sensor Subsystem RV Table. Requirements 1 & 2

| Requirements | Verification |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none">The system must provide battery voltage measurements (± 0.1 V). | <ul style="list-style-type: none">Assemble a CHARM node with a complete sensor subsystem, power delivery subsystem, and system monitor.To directly measure the voltage, place a voltage sensor to measure the battery voltage against ground.Connect the battery to a USB port and begin recording the voltage.To measure the voltage through the battery voltage monitor circuit, read the measurement through the system monitor.Verify the voltage measurement on the monitor is within 0.1 V of the direct voltage measurement. |
| <ul style="list-style-type: none">The system must provide a location within 10 meters of the true location when the device is deployed in an area free of large buildings or other obstacles. | <ul style="list-style-type: none">Assemble a CHARM node with a complete sensor subsystem.Connect to the CHARM node via SSH and run a script to print GPS data.Place a node at a landmark that can be easily identified in satellite imagery, and acquire the landmark's coordinates using Google Maps.Calculate the distance between the GPS coordinates sent by the node and the landmark's coordinates.Verify the above distance is ≤ 10 meters. |

2.4 Routing Subsystem

2.4.1 Overview

The Routing Subsystem is comprised of the Omega2S+ and a few passive components. The Omega2S+ runs processes to route traffic in the mesh, as well as send telemetry about the node state to the System Monitor.

As shown in the Block Diagram (Figure 2), the routing subsystem serves as the center of our design. It takes in data from the GPS Unit and Battery Voltage Monitor via USB and I^2C , respectively, while also receiving and sending network traffic via 2.4 Ghz U.FL antenna. It is powered by the Power Management Subsystem with 3.3 V ($\pm 0.1V$).

This subsystem is responsible for providing WiFi (High-Level Requirement 1) and sending telemetry from the sensors (High-Level Requirement 3).

2.4.2 Requirements

The system satisfies the following requirements:

1. Each node must automatically find and connect with nearby nodes to create a mesh network.
2. Each node must provide a publicly accessible WiFi network with a bandwidth of at least 1 Mbps.
3. Assuming a connection to the internet can be established, each node must be able to send telemetry containing location and battery voltage data at least once per minute.

These requirements were validated with procedures described in Table 3 for Requirement 1, and in Table 4 for Requirements 2 & 3.

2.4.3 Design Decisions

We chose an Omega2S+ for this subsystem as it allows us more flexibility when compared to other options. It is a Linux (OpenWRT) machine, with much more free memory and flash than something like a microcontroller.

The Omega2S+ also comes pre-loaded with various libraries to read from USB and I^2C , ideally making development less bug-prone. The system is very well-documented online, which also factored into our decision, as we want the board's software development process to be as painless as possible.

These benefits come with increased cost. This board is more expensive than potential replacements such as microcontrollers. In the prototyping phase, we value ease of development, good documentation, and software support over cost.

2.4.4 Routing Subsystem RV Table

Table 3: Routing Subsystem RV Table. Requirement 1

| Requirements | Verification |
|-------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none">Each node must automatically find and connect with nearby nodes to create a mesh network. | <ul style="list-style-type: none">Set up a CHARM network with at least two connections from CHARM nodes to the master router, and at least 4 nodes.Power off one of the nodes and record the last time the node sent telemetry.Then, power the node up once again, and verify the node is sending telemetry to the System Monitor. This ensures the node is able to use the mesh for sending network traffic.At a distance of 5 m (± 1 m) from the node, using a smartphone which does not have the CHARM network saved, connect to the network.Connect to the node via SSH and use <code>tcpdump</code> to monitor routed packets.<code>ping</code> any domain on the smartphone, and validate the node routes ICMP packets. |

Table 4: Routing Subsystem RV Table. Requirements 2 & 3

| Requirements | Verification |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> Each node must provide a publicly accessible WiFi network with a bandwidth of at least 1 Mbps. | <ul style="list-style-type: none"> Set up a CHARM network with one node and the master router outside in a large flat area. First, at a distance of 5 m (± 1 m) from the node, using a smartphone which does not have the CHARM network saved, connect to the network. Using Google's online speed test, validate the internet speed is faster than 1 Mbps. Repeat the above at distances in intervals of 5m until 100m, i.e. at 5 m (± 1 m), 10 m (± 1 m), ..., 100 m (± 1 m), and record internet speed as a function of the distance from the node. Plot internet speed versus distance as a line plot, using the line to obtain an estimate of the maximum effective range of the node. Repeat the last three steps with a laptop. |
| <ul style="list-style-type: none"> Assuming a connection to the internet can be established, each node must be able to send telemetry containing location and battery voltage data at least once per minute. | <ul style="list-style-type: none"> Set up a CHARM network with at least two connections from CHARM nodes to the master router, and at least 4 nodes. Record the time. Do nothing for 10 minutes. Power off each node. Analyze the System Monitor telemetry table in its internal database, and validate that all nodes sent consecutive POST requests no longer than 1 minute apart, starting at the recorded time in the previous step. |

2.5 System Monitor Subsystem

2.5.1 Overview

The system monitor is a full-stack web application that displays the location and status of every node. The backend is written in Node.js and includes methods to save and retrieve node data. The frontend is written using React.js and the Google Maps API to visualize the location and status of each node. The frontend design is pictured in Figure 8.

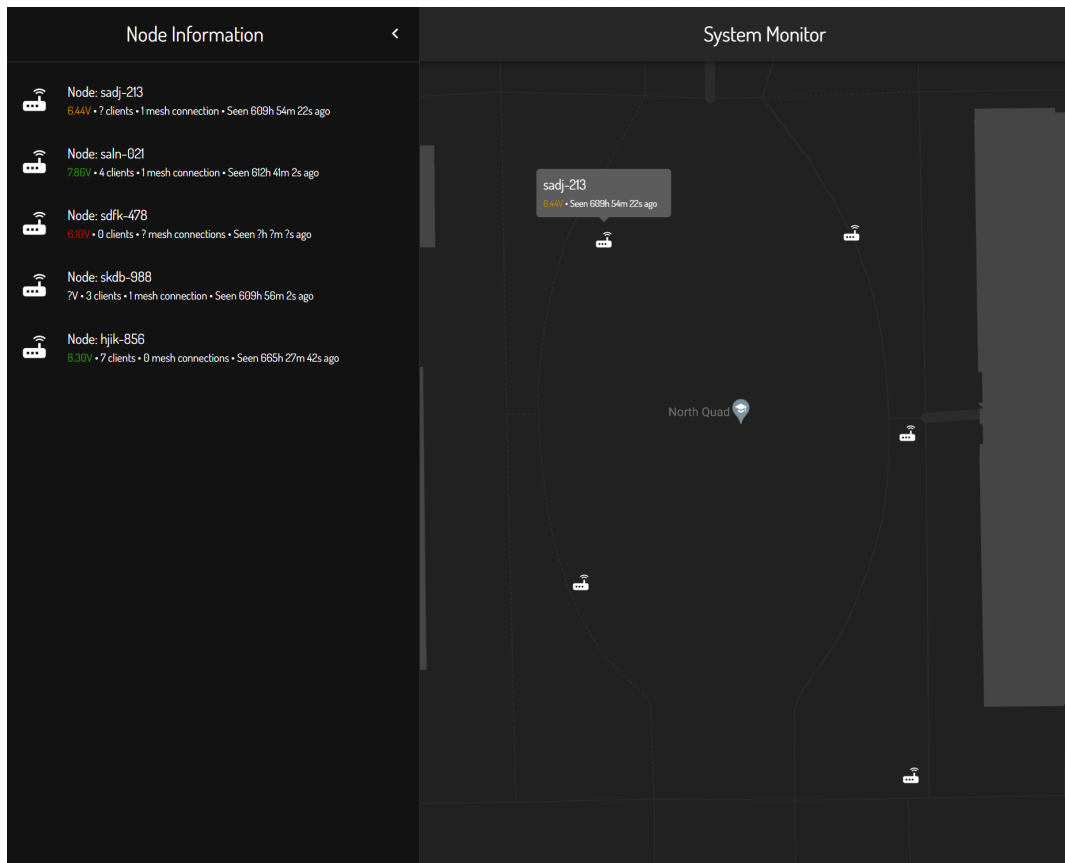


Figure 8: System Monitor Frontend

2.5.2 Requirements

The system satisfies the following requirements:

1. The monitor must be able to display all node locations (± 10 m) on a map.
2. The monitor must show all information on mobile and desktop screens and work with both touchscreen and mouse/keyboard.
3. The monitor must refresh the data at least once per minute.

These requirements were verified using the procedures in Table 5.

2.5.3 Design Decisions

The technologies for the System Monitor (Node.js, React.js) were chosen due to the skillset of our group members. All members of the group have had exposure to Javascript, as well as at least of the frameworks mentioned. This allowed us to implement the system in only 30 hours.

As it stands, the System Monitor is not configured as a production system. The backend is vulnerable to DoS/DDoS attacks, as there is a possibility of logs taking up all disk space on the server in the case of a large influx of HTTP/HTTPS requests. The backend also places too much trust in CHARM nodes, as it only requires a valid API key and CHARM node-id in order to write to the database. This opens the system up to credential-spoofing vulnerabilities. Since this is a proof of concept, these vulnerabilities are mentioned, but not addressed.

2.5.4 System Monitor Subsystem RV Table

Table 5: System Monitor Subsystem RV Table. Requirements 2 & 3

| Requirements | Verification |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none">The monitor must show all information on mobile and desktop screens and work with both touchscreen and mouse/keyboard. | <ul style="list-style-type: none">Open up the System Monitor on a smartphone.Verify that for each node, the Monitor displays the latest telemetry (location and battery voltage).Do the same with the System Monitor opened up on a laptop. |
| <ul style="list-style-type: none">The monitor must refresh the data at least once per minute. | <ul style="list-style-type: none">Set up a CHARM network with at least two nodes.Pick up one of the nodes. Every minute, move the node 30 m in a direction that will still keep the node connected to the other deployed CHARM node.Have someone watching the System Monitor and validating that every move is reflected in the state of the System Monitor. |
| <ul style="list-style-type: none">The monitor must be able to display all node locations on a map. | <ul style="list-style-type: none">Set up a CHARM network with at least two CHARM nodes. Place the nodes on precise landmarks that can be easily identified in satellite imagery.Navigate to the System Monitor web page and get the GPS coordinates for the placed nodes.Calculate the distance between the GPS coordinates sent by the nodes and the true locations of the nodes (precise landmarks on the map).Verify the above to be ≤ 10 m. |

2.6 Physical Design

Our design uses a Polycase WC - 40 polycarbonate enclosure [10]. This enclosure is IP66 certified and is sufficiently dust- and water- resistant for our use case.

2.7 Tolerance Analysis

One of the most important aspects of our design is the battery charging circuit. As suggested by a blog post [5], we utilized the MCP73844 IC to control battery charging. Our schematic closely resembles the reference schematic from the datasheet [6] and can be seen in Figure 6.

Arguably the most important passive component is the shunt resistor (R_4), as this determines the charge current. The datasheet[6] specifically recommends the ERJ-6RQFR22V 220 m Ω , 1%, 1/8 W shunt resistor for ~ 500 mA of charging current. The formula for the resistor value is given in Equation (1).

$$R_{\text{sense}} = \frac{V_{\text{fcs}}}{I_{\text{reg}}} \quad (1)$$

In Equation (1) V_{fcs} is the Fast Charge Current Regulation Threshold, which ranges from 0.1 V to 0.12 V with a nominal value of 0.11 V [6]. I_{reg} is the desired charging current, in amps.

We want 500 mA charging, therefore $R_{\text{sense}} = \frac{0.11}{0.5} = 0.22$ or 220 m Ω . Given the resistor has a 1% tolerance, we can also calculate the minimum and maximum charge current, shown in equations (2) and (3) respectively.

$$\min I_{\text{reg}} = \frac{0.1}{0.2222} = 0.45 = 450\text{mA} \quad (2)$$

$$\max I_{\text{reg}} = \frac{0.12}{0.2178} = 0.551 = 551\text{mA} \quad (3)$$

These are all acceptable currents for our application and meet our design goal of supplying at least 400 mA of peak charging current to our battery cells. Our batteries have 3400 mAh of capacity, so a safe 1C charging speed would be anything up to 3.4 A of current. It is important to limit the current to not overload the USB power source and allow for lower capacity batteries to be safely used, so it makes sense to charge at less than 3.4 A.

The power dissipated over the resistor is given by $P = RI^2$. Given a worst case current of 551 mA, the resistor dissipates up to 66.8 mW of heat, which is within spec of our selected resistor. This calculation matches that made in the datasheet, so their shunt resistor recommendation is acceptable for our use case.

3 Cost Analysis

3.1 Parts/Materials

For this project, we made 5 boards to demonstrate our coverage and meshing capabilities. Refer to Table 8 in Appendix D for all parts used on the PCB board, with their associated costs.

On top of the parts associated with PCB/Notes, we also purchased batteries, enclosure, antenna, and a master gateway router. The total cost adds up to \$856.13.

3.2 Estimated Hours of Development

All of the members in this group are Computer Engineering students. According to Grainger College of Engineering website on post graduate success, the average starting salary for a Computer Engineering graduate is \$105,352 per year [11] which equates to \$52.68 per hour.

Table 6: Estimated Hours

| Category | Approximate Hours | | |
|-----------------------------|-------------------|------------|------------|
| | Martin | Melissa | Trevor |
| Circuit Design | 0 | 0 | 30 |
| Board Layout & Verification | 10 | 20 | 30 |
| Full Stack System Monitor | 30 | 2 | 5 |
| Soldering | 12 | 30 | 6 |
| Prototype & Debugging | 80 | 60 | 75 |
| Documentation & Logistics | 40 | 45 | 30 |
| Total Hours | 172 | 157 | 176 |

Using the hourly estimates in Table 6, we compute the estimated cost for the project labor:

$$\$52.68(\text{Hourly Rate}) * 505(\text{Hours Spent}) * 2.5 = \$66,508.5$$

3.3 External Materials and Resources

- Machine Shop:

The Machine Shop ordered our enclosure and drilled holes for our antennas. This was around **3 hours of work**.

- Senior Design Lab Resources:

We utilized the soldering and testing resources in the Senior Design Lab including soldering materials, a multimeter, and DC power supply.

- Development Resources:

We purchased 2 Onion Omega2 development expansion boards for the Omega2S+. Each of these development boards cost \$15.00 [9].

3.4 Approximate Total Cost

Table 7: Approximate Total Cost

| Category | Estimated Cost |
|-----------------------------|--------------------|
| Material and Parts | \$856.13 |
| Total Labor Cost | \$66,508.5 |
| Development Resources | \$30.00 |
| Total Estimated Cost | \$67,364.63 |

4 Conclusion

4.1 Conclusion

We were able to successfully complete the development of our project with 5 working CHARM nodes, meeting all the requirements previously outlined. During the demonstration, we showcased the connectivity and speed of the CHARM network, the form factor and water resistance of the CHARM nodes, and a functional System Monitor with accurate telemetry.

In the process of development, we discovered several alternative implementations. If we did this project again, we would utilize a consumer router rather than the Omega2S+. There are similarly priced routers that offer faster processing speeds, a 5 GHz radio, USB connectivity, and OpenWRT compatibility. The more powerful wireless access point would have made it easier to implement meshing, and our mesh would have been more performant. We also would have reduced the volume of our nodes, since they contain a lot of empty space.

4.2 Ethics & Safety

Our group acts in accordance with IEEE Code of Ethics adopted by the IEEE Board of Directors through June 2020 [12]. We recognize that technologies have the ability to affect one's life. Thus, we hold ourselves to the highest ethical standard when working professionally in a team. These standards include, but are not limited to:

1. **Seeking and providing truthful reviews and feedback of our technical work [12]**
2. **Constantly learning and acquiring new skills throughout the training and design process [12]**
3. **Treating all people with respect and kindness [12]**

For full details about Ethics and Safety, please refer to Appendix A.

Appendix A Ethics and Safety

A.1 Ethics

Our group acts in accordance with IEEE Code of Ethics adopted by the IEEE Board of Directors through June 2020 [12]. We recognize that technologies have the ability to affect one's life. Thus, we hold ourselves to the highest ethical standard when working professionally in a team which includes but is not limited to:

1. **Seeking and providing truthful reviews and feedback of our technical work [12]**

Within our group, we established a multi-stage review / revision process (Appendix: B) for both software and hardware designs. We will follow course guidelines for timely feedback and confirmation with Teaching Assistant and Professors. On top of existing resources, we also reached out to personal contacts in industry and from extracurricular experiences for design reviews.

Recently, Professor Lumetta has reached out to us and introduced us to Jack Blevins who is a UIUC ECE Alumni with 50+ years of project experience across various industries. Jack is an unbiased reviewer of our technical work, and has already provided valuable insight into potential pitfalls we may run into during the course of the project.

2. **Constantly learning and acquiring new skills throughout the training and design process [12]**

All members have completed the CAD training and will be finishing all training required by the course. We will also consult experienced individuals (Professors, Teaching Assistants, Machine Shop Technicians, etc) if questions or uncertainties arise during this project. This group consists of members with different areas of expertise, including firmware, web development, AI, and robotics. By working together, we will learn from one another.

3. **Treating all people with respect and kindness [12]**

To ensure efficient communication, our team created a Discord server with dedicated channels for subsystems, logistics, and meetings. With a shared Google Drive folder, we make sure that documentation is accessible for all team members. GitHub is used for software and schematic version control. These technologies allow all members to be on the same page with respect to individual responsibilities and overall progress.

A.2 Safety

Here, we consider the potential safety issues relevant to the users of any CHARM network. These include physical safety in the handling of the hardware as well as cyber safety.

A.2.1 Battery Safety

Lithium-ion batteries are dangerous[13][14]. In order to minimize the risk of fire and resultant damage to life and property, we developed a safety manual below with procedures we adhere to when handling the batteries ordered for the project.

Training

- Read, understand, and follow ECE445 guidelines for safe battery usage[15]
- Obtain additional fire safety and fire extinguisher training on top of the required lab training
- Read and understand the datasheets and restrictions for the batteries
- Locate and familiarize themselves with the lab environment and all emergency and safety equipment

Design and Prototype

- Follow the ECE445 Battery Safe Practice Guide[15]
- Ensuring all components selections, circuit designs, and board layout are reviewed and validated by another member and a Teaching Assistant
- Simulating the charging (and any necessary procedures) circuitry before any hardware connection
- Before connecting the battery for the first time, validating and testing the appropriate modules and components if it's possible
- For the first time, only connecting the battery to the circuit in the senior design lab with a Teaching Assistant present and with protective and emergency equipment easily accessible
- Ensuring that the batteries and connected components are stored in a safe place

A.2.2 User Privacy

We do not look at user network packets for any purpose outside of monitoring network load and configuration. If this were a sold product, this would be detailed in a privacy policy.

We recognize the insecure nature of our network. As a publicly accessible network, the system will be vulnerable to bad actors. This is an issue that can be addressed via meetings with cyber security professionals. As this is a proof of concept, this is not something we will be addressing.

A.2.3 System Security

Our System Monitor architecture places trust in CHARM nodes to properly identify themselves with an API key and their node id. This trust can be abused if someone has access to a physical node and is able to hack in and obtain the API key. We do not address this issue in the prototyping phase.

The System Monitor is publicly accessible. A bad actor could get node locations from the website and then steal node(s). This could be mitigated with an authentication system, as well as user permissions that only allow them to see their nodes.

Appendix B Review Processes

These are the processes we are following in the design phase of our project.

B.1 Software

1. Create a branch off of `master` for an assigned task.
2. Develop the code for the feature.
 - (a) Write the code.
 - (b) Ensure proper functionality of the code via unit testing, paying special attention to any relevant edge cases.
3. Create a pull request for the branch, and assign another member of the team to review the written code.
 - (a) The assigned team member should `checkout` the relevant branch.
 - (b) The assigned team member should then ensure proper functionality of the code via unit testing, paying special attention to any relevant edge case.
4. If the pull request is approved, merge the branch into `master`. Otherwise, loop back to step 2.

B.2 Hardware

1. Create a branch off of `master` for an assigned submodule that needs designing.
2. Edit the design in KiCad to include the submodule.
3. Create a pull request for the branch, and assign another member of the team to review the changes to schematic and layout files.
 - (a) The assigned team member should `checkout` the relevant branch.
 - (b) The assigned team member should then validate the design, checking against suggested circuit designs in chip datasheets as well as validating the submodule will meet all of its requirements.
4. If the pull request is approved, merge the branch into `master`. Otherwise, loop back to step 2.

Appendix C Soldering Verification

Follow the instructions in this appendix to validate soldered subsystems.

C.1 USB-C Port

1. Plug a USB-C Charger into the USB-C port.
2. Ensure the Red LED (D1) lights up.

C.2 Boost Converter

1. Solder and validate previous subsystems.
2. Plug a USB-C Charger into the USB-C port.
3. Ensure there are 12 V across the 1 mF Capacitor (C9).

C.3 Buck Converter

1. Solder and validate previous subsystems.
2. Ensure the Switch is in OFF position.
3. Hook up a power supply to supply V_{batt} .
4. Set power supply to 7 V.
5. Flip the Switch to ON.
6. Ensure voltage across 220 uF Capacitors (C4 and C5) is 3.3 V.
7. Ensure Red LED D2 is on.

C.4 Charge Controller

1. Solder and validate previous subsystems.
2. Ensure the Switch is in OFF position.
3. Measure V_{batt} .
4. Plug a USB-C Charger into the USB-C port.
5. Wait 5 minutes.
6. Unplug the USB-C Charger from the USB-C port.
7. Measure V_{batt} .
8. Ensure the second V_{batt} measurement is higher than the first.

C.5 Omega 2S+

1. Solder and validate previous subsystems.
2. Insert two batteries into the PCB battery holder, ensuring switch is in the "off" position.
3. Flip the switch.
4. The Omega2S+ LED should light up, begin flashing, and then eventually remain solid.
5. Once the LED remains solid (should be around 1 minute), check the WiFi networks in the lab. There should be a WiFi network with SSID `Omega-XXXX` where `XXXX` are the last two bytes of the Omega2S+'s MAC address, in hexadecimal notation.

C.6 Battery Monitor

1. Solder and validate previous subsystems.
2. Connect to the Omega2S+'s WiFi network.
3. SSH into the Omega2S+ via the `ssh root@omega-XXXX.local` command.
4. Enter the default password when prompted: `onioneer`.
5. Download our script for testing the ADC:
`wget https://raw.githubusercontent.com/OnionIoT/adc-exp/master/adc-exp.sh`
6. Run the script: `ash adc-exp.sh 0`.
7. Measure the voltage at the ADC's point of measurement and validate that it roughly matches the voltage output by the script.

C.7 GPS Sensor

1. Solder and validate previous subsystems.
2. Connect to the Omega2S+'s WiFi network.
3. SSH into the Omega2S+ via the `ssh root@omega-XXXX.local` command.
4. Enter the default password when prompted: `onioneer`.
5. Run the following command: `cat /dev/ttyACM0`.
6. You should see the output similar to that pictured in Figure 9.

```
Command Prompt - ssh root@omega-a390.local
root@Omega-A390:~# cat /dev/ttyACM0
$GNRMC,,V,,,,,,,,,N,V*37

$GNVTG,,,,,,,,N*2E

$GNGGA,,,,,0,00,99.99,,,,,*56

$GNGSA,A,1,,,,,,,,,99.99,99.99,99.99,1*33

$GNGSA,A,1,,,,,,,,,99.99,99.99,99.99,2*30

$GNGSA,A,1,,,,,,,,,99.99,99.99,99.99,3*31

$GNGSA,A,1,,,,,,,,,99.99,99.99,99.99,4*36

$GPGSV,1,1,00,1*64

$GLGSV,1,1,00,1*78

$GAGSV,1,1,00,7*73

$GBGSV,1,1,00,1*76

$GNGLL,,,,,V,N*7A

$GNRMC,,V,,,,,,,,,N,V*37

$GNVTG,,,,,,,,N*2E

$GNGGA,,,,,0,00,99.99,,,,,*56
```

Figure 9: Expected cat /dev/ttyACM0 command output

Appendix D PCB Parts & Cost

Table 8: PCB Part Cost Analysis

| Description | Part Number | Unit Price | #/Board | \$/Board | Cost (5 Boards) |
|-------------------------------------|---------------------|------------|---------|-----------------|-----------------|
| Battery Charge IC | MCP73844-840I/MS | \$1.78 | 1 | \$1.78 | \$8.90 |
| Battery Holder | 1123 | \$5.69 | 1 | \$5.69 | \$28.45 |
| MOSFET | IRF7404TRPBF | \$1.11 | 1 | \$1.11 | \$5.55 |
| 220m Resistor | ERJ-6RQFR22V | \$0.35 | 1 | \$0.35 | \$1.75 |
| On/Off Switch | L101011MS02Q | \$2.26 | 1 | \$2.26 | \$11.30 |
| 50k Ω Resistor | RT0603BRD0750KL | \$0.33 | 2 | \$0.66 | \$1.65 |
| 0.1uF Capacitor | T491A104K035AT | \$0.62 | 3 | \$1.86 | \$3.10 |
| 10uF Capacitor | EEE-FN1E100R | \$0.50 | 3 | \$1.50 | \$2.50 |
| 20k Ω Resistor | RT0603DRE0720KL | \$0.12 | 1 | \$0.12 | \$0.60 |
| 4.7k Ω Resistor | RT0603DRE074K7L | \$0.12 | 2 | \$0.24 | \$0.60 |
| ADC | ADS1113IDGST | \$6.49 | 1 | \$6.49 | \$32.45 |
| 330nF Capacitor | ECPU1C334MA5 | \$0.88 | 1 | \$0.88 | \$4.40 |
| 1mF Capacitor | 16SVPF1000M | \$2.52 | 1 | \$2.52 | \$12.60 |
| Schottky Diode | SS24FL | \$0.45 | 1 | \$0.45 | \$2.25 |
| Boost Converter IC | LM2585S-12/NOPB | \$10.41 | 1 | \$10.41 | \$52.05 |
| 68uH Inductor | MSS1210-683MED | \$2.42 | 1 | \$2.42 | \$12.10 |
| 2.94k Resistor | RC0603FR-072K94L | \$0.10 | 1 | \$0.10 | \$0.50 |
| 100uF Capacitor | 16SVPC100M | \$1.19 | 1 | \$1.19 | \$5.95 |
| Schottky Diodes & Rectifiers 20V 5A | B520C-13-F | \$0.92 | 1 | \$0.92 | \$4.60 |
| Buck Converter IC | LM2576SX-3.3/NOPB | \$4.06 | 1 | \$4.06 | \$20.30 |
| 100uH Inductor | 7.44374E+13 | \$9.43 | 1 | \$9.43 | \$47.15 |
| 220uF Capacitor | UUD1C221MCL1GS | \$0.40 | 3 | \$1.20 | \$2.00 |
| 100nF Capacitor | C1005X7R1C104K050BC | \$0.10 | 1 | \$0.10 | \$0.50 |
| Yellow LED | 150080YS75000 | \$0.19 | 1 | \$0.19 | \$0.95 |
| GPS IC | NEO-M9N-00B | \$27.00 | 1 | \$27.00 | \$135.00 |
| U.FL Connector | U.FL-R-SMT(01) | \$1.35 | 1 | \$1.35 | \$6.75 |
| 27nH Inductor | LQW18AN27NG00D | \$0.19 | 1 | \$0.19 | \$0.95 |
| 10 Ω Resistor | RT1206FRE1310RL | \$0.14 | 1 | \$0.14 | \$0.70 |
| 27 Ω Resistor | RR0816Q-270-D | \$0.13 | 4 | \$0.52 | \$0.65 |
| Schottky Diode | 1N5819 | \$0.40 | 1 | \$0.40 | \$2.00 |
| Green LED | 150080VS75000 | \$0.19 | 1 | \$0.19 | \$0.95 |
| 50 Ω Resistor | RT0603BRE0750RL | \$0.49 | 1 | \$0.49 | \$2.45 |
| Omega2S+ | OM-O2SP | \$24.12 | 1 | \$24.12 | \$120.60 |
| 2.5A Fuse | SF-1206F250-2 | \$0.60 | 1 | \$0.60 | \$3.00 |
| USB-C Port | USB4135-GF-A | \$0.75 | 1 | \$0.75 | \$3.75 |
| 0 Ω Resistor | CRCW06030000Z0EAHP | \$0.15 | 1 | \$0.15 | \$0.75 |
| 5.1k Resistor | RT0603DRE075K1L | \$0.13 | 2 | \$0.26 | \$0.65 |
| Red LED | 150080RS75000 | \$0.19 | 3 | \$0.57 | \$0.95 |
| 1k Ω Resistor | RT0603FRE131KL | \$0.10 | 4 | \$0.40 | \$0.50 |
| Total | NA | NA | NA | \$113.06 | \$541.85 |

References

- [1] J. DeYoung. "When disaster strikes: Decentralized tech a safety mesh from blackouts." (2020), [Online]. Available: <https://cointelegraph.com/news/when-disaster-strikes-decentralized-tech-a-safety-mesh-from-blackouts> (visited on 07/11/2020).
- [2] M. Hamblen. "'Massive damage' from Japan quake hits communications." (2011), [Online]. Available: <https://www.computerworld.com/article/2506865/-massive-damage--from-japan-quake-hits-communications.html> (visited on 07/11/2020).
- [3] U. Ashraf, A. Khwaja, J. Qadir, S. Avallone, and C. Yuen, "Wimesh: Leveraging mesh networking for disaster communication in poor regions of the world," *CoRR*, vol. abs/2101.00573, 2021. arXiv: 2101.00573. [Online]. Available: <https://arxiv.org/abs/2101.00573>.
- [4] T. Instruments. "WEBENCH Power Designer." (2022), [Online]. Available: <https://www.ti.com/design-resources/design-tools-simulation/webench-power-designer.html> (visited on 07/11/2020).
- [5] S. Bhowmick. "Designing an Advanced 2S Li-Ion/ Li-Po Battery Charge System using MCP73844 IC." (2021), [Online]. Available: <https://circuitdigest.com/electronic-circuits/designing-an-advance-2s-li-ion-li-po-battery-charger-system-using-mcp73844-ic> (visited on 07/11/2020).
- [6] "Advanced Single or Dual Cell Lithium-Ion/ Lithium-Polymer Charge Management Controllers", MCP73841/2/3/4, Microchip, Nov. 2012.
- [7] "NEO-M9n-00B ", UBX-19014285, u-blox, Jun. 2021. [Online]. Available: <https://www.u-blox.com/en/docs/UBX-19014285>.
- [8] "NEO-M9n-00B Integration Manual ", UBX-19014286, u-blox, Jun. 2021. [Online]. Available: https://content.u-blox.com/sites/default/files/NEO-M9N_Integrationmanual_UBX-19014286.pdf.
- [9] Onion. "'Onion Doc'." (2022), [Online]. Available: <https://docs.onion.io/omega2-docs/expansion-dock.html> (visited on 09/29/2022).
- [10] "WC-40 Outdoor Enclosure with Clear Cover", WC-34, Polycase, Mar. 2011. [Online]. Available: <https://www.polycase.com/wc-40>.
- [11] T. G. C. of Engineering. "'The Grainger College of Engineering - Computer Engineering'." (2022), [Online]. Available: <https://grainger.illinois.edu/academics/undergraduate/majors-and-minors/computer-engineering#:~:text=Post%5C%2DGraduation%5C%20Success&text=The%5C%20average%5C%20salary%5C%20between%5C%202020,median%5C%20signing%5C%20bonus%5C%20of%5C%20%5C%2415%5C%2C000> (visited on 09/29/2022).
- [12] IEEE. "'IEEE Code of Ethics'." (2016), [Online]. Available: <https://www.ieee.org/about/corporate/governance/p7-8.html> (visited on 02/08/2020).
- [13] A. Semuels. "When Your Amazon Purchase Explodes." (2019), [Online]. Available: <https://www.theatlantic.com/technology/archive/2019/04/lithium-ion-batteries-amazon-are-exploding/587005/> (visited on 09/29/2022).
- [14] J. Jhaveri. "Battery Safety : Top 5 Reasons Why Lithium-Ion Batteries Catch Fire." (2022), [Online]. Available: <https://www.ionenergy.co/resources/blogs/battery-safety/> (visited on 09/29/2022).

- [15] *"Safe Practice for Lead Acid and Lithium Batteries"*, General Battery, ECE445 Senior Design Spring 2016 Course Staff, Jan. 2016. [Online]. Available: <https://courses.grainger.illinois.edu/ece445/documents/GeneralBatterySafety.pdf>.